Express Mail Label No.: EL532334701US
Date of Deposit: 12th October 1999

PERFORMANCE TESTING OF SERVER SYSTEMS

Field of the Invention

This invention relates to client-server computing environments, in which one or more server machines execute requests issued by, typically, a large number of client machines. The invention relates particularly to performance testing of servers for the purpose of determining whether design and/or operational criteria are met. This leads to a determination of the adequacy of sizing of a server.

Background of the Invention

In modern scalable computing systems a common topology has three (logical and/or physical) tiers: (i) a presentation tier characterised by multiple workstations focusing on user interactions, (ii) a business tier characterised by multiple servers executing application/business logic, (iii) a data tier characterised by multiple databases working on data storage and organization. The physical systems are interconnected by a communications network, examples being Local or Wide Area Networks (LAN/WAN).

Such computing systems find application in many and varied fields, ranging from university research and teaching facilities to business applications. In fact, almost every business will utilise such a system to transact its functions and serve its clients. For example, a system may be used to control inventory, for image processing and accounts purposes, and for servicing client's enquiries. Many businesses have very large client bases and may provide an extensive inventory of goods and services. One illustrative example is a telecommunications service provider (Telco) that serves a countrywide client base. The Telco's subscribers thus can number in the millions, and each customer will expect a near immediate response from a Customer Service Representative (CSR) to any inquiry, which can range from billing information, a request for a new service, or the placing of orders for a product.

Similar examples are seen in Utilities, insurance companies, banks, hospitals, law firms, accountancy firms, stock exchanges, universities and Government agencies, to name but a few.

In the course of developing large-scale client server computing systems, an important part of the design process is to determine whether performance criteria such as (i) the average response time of a nominated transaction, and (ii) the proportion of CPU time (Client, Server or Database server) taken by a nominated transaction, are met. These determinations can lead to the conclusion that the computing hardware is correctly sized.

A known technique of performance testing is termed 'stress testing' or 'Benchmarking', by which simulated transaction records are 'fed' to the server computer, and as that loading is increased, performance criteria are measured.

Two specific examples of stress testing known in the prior art are disclosed in Published Japanese Application No. 10-187495 (NEC Corp), entitled "Method and Device for Evaluating High-load Emulation Performance", and in US Patent No. 5,790,425 (Wagle, assigned to Sun Microsystems, Inc.), issued on August 4, 1998, entitled "Generic Server Benchmarking Framework in Client Server Environment". Both of these prior art documents offer only an approximation of actual loading due to execution of the live application.

It is an object of the invention to at least address this shortcoming.

Summary of the Invention

The invention provides a method for testing server performance, comprising the steps of:

(a) forming a collection of live maps for a plurality of transactions for a chosen computing application;

(b) transmitting a processing load, constituted by a plurality of said maps for a plurality of said transactions, to a server running said computing application; and

(c) measuring one or more performance criteria for said server as it executes said load.

The invention further provides a method for testing server performance, comprising the steps of:

(a) forming a collection of live maps for a plurality of transactions for a chosen computing application;

(b) transmitting a processing load, constituted by a plurality of said maps for a plurality of transactions, from a workstation to a server running said computing application;

(c) for each transaction within said load, returning a result to said workstation; and

(d) measuring, at said workstation, one or more performance criteria based on execution of said load by said server.

The processing load can be varied by making changes to the number of maps and the mix of transactions transmitted to the server. The measurements of the performance criteria will be repeated for each individual processing load. The measured performance criteria can be compared against predetermined performance measures to determine whether the server's capacity is satisfactory. The performance criteria can include the average response time for a transaction within a load, and the proportion of the server CPU time taken by each transaction of the load. The performance criteria can be compared against predetermined stored performance measures to determine whether server capacity is satisfactory. The performance criteria measurement can be performed on the workstation, as opposed to the server. Further, the server can have connection to one or more database servers that execute portions of the load transactions. The performance criteria can be end-to-end, namely from workstation to server to database server.

Brief Description of the Drawings

Embodiments of the invention will now be described with reference to the accompanying drawings, in which:

Fig. 1 is a representative topology of a three tier computing system;

Fig. 2 is a generalised software architecture for a client-server environment;

Fig. 3 shows a representative transport layer package passed between client and server;

Figs. 4a and 4b show topographies of stress testing systems; and

Fig. 5 shows the software elements created to implement performance testing.

Description of Preferred Embodiments and Best Mode

Fig. 1 is a representative topology of a three tier computing system 10 embodying the invention. The presentation (or client/user) tier is represented by a number (1...n) of workstations 20, that can be appropriate computing terminals, for example personal computers. The business tier is represented by a number (1...p) of servers 30, that can be dedicated mini or mainframe computers. The data tier is represented by a number (1...m) of database servers 40, which can include dynamically managed magnetic or optical storage media.

The computing system 10 is of an 'open' design, providing communication links 60,62,64, via external networks 70,72,74 to like-devices 22,32,42 and remote telephone terminals 24, 26.

The workstations 20, servers 30, and databases 40 are interconnected by a Local or Wide Area Network (LAN or WAN) 50. The LAN/WAN 50 carries information passing between each of the three basic elements described.

Client/Server systems such as shown in Fig. 1 find industrial application in the fields noted in the foregoing Background section. For the purposes of a non-limiting illustration, consider the example of a Telco operating across many States of the United States. Such a Telco will typically support local, regional, interstate and international voice and data calls, as well as cellular mobile voice and data traffic. Customers of the Telco can choose from a wide range of goods and services including, for example, the installation of second phone/fax/Internet lines, call forwarding, and messaging. They also will expect to be able to make enquiries of CSRs stationed at the workstations 20 concerning billing and service faults. It is not unreasonable to expect a modern-day Telco to have at least 1 million customers, typically requiring at least 500 CSRs. A Telco system infrastructure of this size can expect to handle about 15,000 business transactions per hour. Depending on the business function being used, the CSR will interact with the system one or more times. Each client/server interaction may require few to many database interactions (reading or writing to the physical database).

To give a better example of the size of computing hardware required to achieve such performance, the CSR workstations 20 could be Pentium™ personal computers running the Windows NT™ operating system, the servers 30 can be one or more IBM UNIX™-based 12-way RS6000™ S-70 machines, and the databases would require a

capacity of about 40 Gbytes, managed by an Oracle ™ or IBM DB-2™ system. There would, of course, be other operational LAN/WAN servers required to handle data communications, as would be readily understood by a person skilled in the art.

Because of the very large hardware commitment, and expense, in such client/server systems, it is important that the correct sizing is achieved, in the sense that the hardware is neither two large nor too small to achieve the desired performance characteristics.

Fig. 2 is a generalised software architecture for a client-server environment. On the client machine, a Graphical User Interface (GUI) layer provides the human-machine interface for a user. The GUI layer interfaces with an application layer, where the specific computing operation or purpose performed by the client-server system resides. The application layer interfaces with a middleware layer that handles system aspects such as system resource usage, operating system locks, shared memory access, container services, queuing services, transaction services, logical unit of work coordination, inter-process communications, user access control services and configuration retrieval services. As shown, application data, packaged into "maps" or "containers", is passed to the middleware layer. The middleware layer represents the operating system and communications services. The transport layer of the client machine is in network communication with the server machine. The server machine replicates the transport layer, the middleware layer, and the application layer functions.

The content of a map/container includes the identification of the 'service' which the server machine application is to execute, together with the application data which is required by the particular application process. Fig. 3 shows a representative data packet having header information specific to the transport and middleware layers. Optionally, there can be similar trailer information. The map/container comprises the services information and application data.

For a computing system as shown in Fig. 1, there can be many and varied configurations, however it is common for there to be a large number of client workstations 20, loading one or more application servers 30. In the peformance (or stress) testing environment, it is common for the plurality of client machines to be emulated by a single larger-scale server machine.

Sub A3

Fig. 4a shows an example of a server machine 100, emulating a client machine, in networked connection with a server machine 102 that is to stress-tested.

Fig. 4b shows the same server machine 100 emulating a client machine, however the 'server' to be tested includes a front-end application server 104 having connection to a plurality of database servers 106, in turn connected with data stores 108. The method of the invention is applicable to the arrangement of Figs. 4a and Fig. 4b, and other variations.

The methodology of the service performance testing includes the following (non-limiting) broad steps:

Sub A4

(i) The of live maps/containers for a plurality of transactions for a chosen application must firstly be collected. By "live" is meant actual transactions, as opposed to simulations.

(ii) The collection of containers is stored within the client emulation server.

(iii) A processing load is transmitted from the emulation server to the server under test, and the selected performance criteria are measured as the server executes the load.

(iv) The processing load is varied, both in terms of the total number of transactions and the transaction type (or mix), that is transmitted to the server.

(v) The performance criteria can be utilised to determine whether the sizing of the server meets intended design parameters.

Fig. 5 shows the software elements that are created to implement performance testing in the terms described above. The files are delimited by those created in advance of the performance testing (i.e. pre-runtime), represented by numeral 120, and those elements that are utilised in the course of the performance testing, represented by the numeral 122.

Sub A5

In the pre-runtime, a Business Workload Definition File is created and populated. This file and a mapping file (mapping Business Transactions to Machine Transactions) are merged to create the machine workload, resulting in a Machine Workload Execution Definition File. In the run-time, the pre-stored live maps are selectively read by a map sending program which executes the Workload Execution File to place the process load onto the server 102 running the application under test. The Map Sending Program is replicated: one per client machine being simulated. The server 102 under test executes the requested load and returns a reply map. Such reply maps are stored on the emulated client machine in the Maps Received File. It is necessary for the Business Workload Definition File and the Mapping File to relate to the same application that is being run by the server 102 under test. In the same way, the stored maps must relate to the same server application.

Sub A6

The performance criteria, such as the average response time of a transaction or the proportion of CPU time taken by a transaction, can be determined by the server under test itself, or can be determined on the client emulation server (to include the communications link performance). Whichever way, the results of the performance testing are stored in a Logging File on the client emulation server or on the server under test.

Sub A7

An example of the Business Workload Definition File, for a Telco customer enquiry and ordering system (such as generally described above) is as follows:

	EQ	**79**	**Enquiries**
	EA	21	Account enquiries
	ES	10	Statement enquiries
	EG	21	General enquiries
	ET	34	Toll enquires
	EL	14	Calling card

The first line represents that, of the total workload, 79% is occupied by "Enquiries". The following rows specify the sub-type of enquiries within that 79%. For example, an Account enquiry represents 21% of the total enquiries, while the total enquiries are 79% of the total workload.

Sub A8

An example of the file which maps Business Transactions (of sub-type DA) to a sequence of maps to be executed is as follows:

```
*       The Master Workload Detail file

*SubTyp DA                  (The particular subtype being defined
*       (A sequence of individual maps to execute
*       vgrous03
*       vgrous04
*       vgrprd06
*       vgprd06
*       vgracc01
*       vgracc03
[vgracc63, 1; vgracc61,1; vgracc53,1;]  (Name, relative probability
```

Sub A9

An example of Machine Workload Execution definition file is as follows:

```
*Execution Script for build56. Script. T1
*       Subtype = EA
VGRACC38   VC38   I.VGRACC38.  XXX. 060
VGRACCNO VCNO I.VGRACCNO.XXX. 035
VGRPRPDF   VTDP  I.VGRPRPDF.  XXX. 005
VGRPRP06   VP06  I.VGRPRD06.  XXX. 064
VGRACC01   VC01  I.VGRACC01.   XXX. 068
VGRACC65   VC65  I.VGRACC65.   XXX. 026
*       Subytpe = EA
VGRACC38   VC38. I.VGRACC38.   XXX 060
VGRACCNO VCNOI.VGRACCNO. XXX. 004
VGRPRPDF   VTPD I.VGRPRPDF.   XXX 065
VGRPRD06   VT06  I.VGRPRD06.   XXX. 015
VGRACC01    VC0   I.VGRACC01.   XXX. 042
VGRACC69    VC69 I.VGRACC69.   XXX. 032
*       Subtype = EG
VGRACC38    VC38. I.VGRACC38.  XXX. 003
VGRACCNO  VCNOI.VGRACCNO. XXX. 013
```

VGRPRPF VTPD I. VGRPRPF. XXX. 116
VGRPRD06 VT06 I.VGRPRD06. XXX. 069
VGRACC01 VC01 I.VGRACC01. XXX. 096

The third field is the name of the specific map file.

Example

Sub A10

Referring again to Fig. 2, examples of implementations for the middleware layers include the IBM CICS™ or ENCNIA™ systems. In relation to the transport layer, examples of implementations are either TCP/IP or SNA. Any convenient physical layer network can be utilised, such as a token passing LAN. The application layer must have the capability, either inherently or by specific coding, to create or write live maps.

The measurements shown below were performed on a single node (model 595) of an RS/6000 SP 2 system.

The Business Workload Distribution file was of a similar composition to that shown above. The client emulating server machine also was an RS/6000 machine. The performance metric was to determine the maximum CICS throughput rate for the specified enquiry workload. Workload was increased in the increments of two, three, four and six simulated terminals, with the response time being calculated for each transaction.

The following table represents the individual transactions for the case of "end time", the second column represents the discrete individual "transactions", the third column shows the "start time", and the fourth column shows the overall response time.

11/26/98, 15:23:01, i. VGRACCNO.xxx.059, 15:23:00, 0.94499345
11/26/98, 15:23:02, i. VGRPRPDF .xxx. 065, 15:23:01, 1.52325305
11/26/98, 15:23:03, i.VGRPRD06. xxx. 007, 15:23:02, 0.73049395
11/26/98, 15:23:04, i.VGRPRD06. xxx. 091, 15:23:03, 1.096042
11/26/98, 15:23:07, i.VGRACC01. xxx. 042, 15:23:04, 3.0945521
11/26/98, 15:23:09, i.VGRACC05. xxx. 019, 15:23:07, 2.28059385
11/26/98, 15:23:13, i.VGRACC38. xxx. 012, 15:23:09, 3.57596095

11/26/98, 15:23:14, i.VGRACCNO.xxx.114, 15:23:13, 0.59853705
11/26/98, 15:23:15, i.VGRPRPDF.xxx.005, 15:23:14, 1.61760075

.
.
.
.

11/26/98, 15:28:34, i.VGRACCNO.xxx.013, 15:28:34, 0.4899564
11/26/98, 15:28:34, i.VGRPRPDF.xxx.014, 15:28:34, 0.43951875
11/26/98, 15:28:35, i.VGRPRD06.xxx.064, 15:28:35, 0.33546205
11/26/98, 15:28:35, i.VGRPRD06.xxx.007, 15:28:35, 0.41166125
11/26/98, 15:28:37, i.VGRACC01.xxx.042, 15:28:35, 1.8305234
11/26/98, 15:28:38, i.VGRACC05.xxx.098, 15:28:37, 1.0756061
11/26/98, 15:28:40, i.VGRACC38.xxx.087, 15:28:38, 1.6714174
11/26/98, 15:28:40, i.VGRACCNO.xxx.013, 15:28:40, 0.298258
11/26/98, 15:28:41, i.VGRPRPDF.xxx.065, 15:28:40, 0.94981075
11/26/98, 15:28:42, i.VGRPRD06.xxx.015, 15:28:41, 0.5698334
11/26/98, 15:28:44, i.VGRACC01.xxx.042, 15:28:42, 2.63401085
11/26/98, 15:28:46, i.VGRACC38.xxx.060, 15:28:44, 1.13616375
11/26/98, 15:28:46, i.VGRACCNO.xxx.013, 15:28:46, 0.4442817
11/26/98, 15:28:47, i.VGRPRPDF.xxx.065, 15:28:46, 0.7981063
11/26/98, 15:28:47, i.VGRPRD06.xxx.091, 15:28:47, 0.4851278
11/26/98, 15:28:48, i.VGRPRD06.xxx.069, 15:28:47, 0.49962255
11/26/98, 15:28:49, i.VGRACC01.xxx.068, 15:28:48, 1.5193212
11/26/98, 15:28:51, i.VGRACC05.xxx.019, 15:28:49, 1.1684261
11/26/98, 15:28:52, i.VGRACC38.xxx.012, 15:28:51, 1.72167155
11/26/98, 15:28:53, i.VGRACCNO.xxx.059, 15:28:52, 0.62635305
11/26/98, 15:28:55, i.VGRPRPDF.xxx.014, 15:28:53, 2.46022115
11/26/98, 15:28:56, i.VGRPRD06.xxx.007, 15:28:55, 0.3547103
11/26/98, 15:28:57, i.VGRACC01.xxx.016, 15:28:56, 1.07111495
11/26/98, 15:28:58, i.VGRACC63.xxx.110, 15:28:57, 0.7502934
11/26/98, 15:28:59, i.VGRACC38.xxx.087, 15:28:58, 1.04842535
11/26/98, 15:28:59, i.VGRACCNO.xxx.029, 15:28:59, 0.444598
11/26/98, 15:29:00, i.VGRPRPDF.xxx.005, 15:28:59, 0.6602939
11/26/98, 15:29:00, i.VGRPRD06.xxx.064, 15:29:00, 0.3538677
11/26/98, 15:29:01, i.VGRACC01.xxx.096, 15:29:00, 1.05042975

The following table summarises the performance testing, where the first column represents the number of terminals, and the second column represents the number of transactions per second.

Terminals	Trans /sec	Comments
2	5.8	
3	6.1	
4	7.2	
6	4.8	Blocking on I/O write
4	9	
11	7.75	Blocking again

When the number of terminals is increased to six, the reduction in the throughput indicated that there was blocking on the I/O writing, and an appropriate adjustment was made, namely the parameter 'CisTimeMode' was set to 0. With this change made, four terminals were simulated, then eleven. The reduction in the number of transactions per second indicates the existence of another bottleneck. This led to the suggestion that there is insufficient memory on the server machine to handle the load generated by eleven client machines.

The example presented increased the number of terminals, while maintaining the Workload Execution Definition file as constant. It is equally possible to hold the number of terminals fixed and increase the number and mix of transactions.

One advantage of the invention is that the GUI layer (see Fig. 2) format can be changed and yet there would be no requirement to re-record the set of live maps.

It will be understood that the scope of the invention fully encompasses other embodiments which may become obvious to those skilled in the art, and that the scope of the present invention is accordingly to be limited by nothing other than the appended claims.